

LEETCODE QUESTIONS

leetcode.com/problems/linked-list-cycle/

Problem List < > Submit

Description Accepted Editorial Solutions Submissions

All Submissions

Accepted 29 / 29 testcases passed
Puneeth_Jagadeesha submitted at Dec 19, 2025 22:47

Runtime 7 ms | Beats 87.70%
Memory 11.23 MB | Beats 38.86%

Bar chart showing runtime distribution for the linked-list-cycle problem. The x-axis represents runtime in milliseconds (3ms, 5ms, 7ms, 9ms, 11ms, 13ms). The y-axis represents the percentage of submissions (0%, 10%, 20%, 30%). The distribution shows a peak at 13ms, with approximately 20% of submissions running in that time range.

Code | C

```
1 /**
2  * Definition for singly-linked list.
3  * struct ListNode {
4  *     int val;
5  *     struct ListNode *next;
6  * };
7  */
8 bool hasCycle(struct ListNode *head) {
```

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

head =
[3,2,0,-4]

pos =
1

leetcode.com/problems/merge-two-binary-trees/submissions/1859978213/

Problem List < > Submit

Description Accepted Editorial Solutions Submissions

All Submissions

Accepted 182 / 182 testcases passed
Puneeth_Jagadeesha submitted at Dec 19, 2025 22:49

Runtime 0 ms | Beats 100.00%
Memory 18.88 MB | Beats 64.00%

Bar chart showing runtime distribution for the merge-two-binary-trees problem. The x-axis represents runtime in milliseconds (1ms, 2ms, 3ms, 4ms, 5ms, 6ms). The y-axis represents the percentage of submissions (0%, 25%, 50%, 75%, 100%). The distribution shows a peak at 1ms, with approximately 75% of submissions running in that time range.

Code | C

```
1 /**
2  * Definition for a binary tree node.
3  * struct TreeNode {
4  *     int val;
5  *     struct TreeNode *left;
6  *     struct TreeNode *right;
7  * };
8  */
```

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

root1 =
[1,3,2,5]

root2 =
[2,1,3,null,4,null,7]

leetcode.com/problems/middle-of-the-linked-list/submissions/1859979958/

Problem List < > Submit

Description Accepted x Editorial Solutions Submissions

All Submissions

Accepted 36 / 36 testcases passed
Puneeth_Jagadeesha submitted at Dec 19, 2025 22:51

Runtime 0 ms Beats 100.00% Memory 8.36 MB Beats 93.45%

Analyze Complexity

Code: C

```
1 /**
2  * Definition for singly-linked list.
3  * struct ListNode {
4  *     int val;
5  *     struct ListNode *next;
6  * };
7  */
8 struct ListNode* middleNode(struct ListNode* head) {
9     struct ListNode *slow = head;
10    struct ListNode *fast = head;
11
12    while (fast != NULL && fast->next != NULL) {
13        slow = slow->next;
14        fast = fast->next->next;
15    }
16    return slow;
17 }
```

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

head =
[1,2,3,4,5]

Output

[3,4,5]

leetcode.com/problems/remove-linked-list-elements/submissions/1859980546/

Problem List < > Submit

Description Accepted x Editorial Solutions Submissions

All Submissions

Accepted 66 / 66 testcases passed
Puneeth_Jagadeesha submitted at Dec 19, 2025 22:52

Runtime 0 ms Beats 100.00% Memory 12.48 MB Beats 78.76%

Analyze Complexity

Code: C

```
1 /**
2  * Definition for singly-linked list.
3  * struct ListNode {
4  *     int val;
5  *     struct ListNode *next;
6  * };
7  */
8 struct ListNode* removeElements(struct ListNode* head, int val) {
9     while (head != NULL && head->val == val) {
10         head = head->next;
11     }
12
13     struct ListNode *curr = head;
14
15     while (curr != NULL && curr->next != NULL) {
16         if (curr->next->val == val) {
17             curr->next = curr->next->next;
18         } else {
19             curr = curr->next;
20         }
21     }
22     return head;
23 }
```

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

head =
[1,2,6,3,4,5,6]

val =
6